SAVE POINTS

Before knowing about the save points let us discuss about the commit and rollback operations

COMMIT: If everything is in order with all statements within a single transaction, all changes are recorded together in the database is called **committed**. The COMMIT command saves all the transactions to the database since the last COMMIT or ROLLBACK command.

Syntax:

COMMIT;

Example: Sample table 1

Student				
Rol_No	Name	Address	Phone	Age
1	Ram	Delhi	9455123451	18
2	Ramesh	Gurgaon	9652431543	18
3	Sujit	Rohtak	9156253131	20
4	Suresh	Delhi	9156768971	18
3	Sujit	Rohtak	9156253131	20
2	Ramesh	Gurgaon	9652431543	18

Following is an example which would delete those records from the table which have age = 20 and then COMMIT the changes in the database.

Queries:

```
DELETE FROM Student WHERE AGE = 20;
COMMIT;
```

Output:

Thus, two rows from the table would be deleted and the SELECT statement would look like,

Rol_No	Name	Address	Phone	Age
1	Ram	Delhi	9455123451	18
2	Ramesh	Gurgaon	9652431543	18
4	Suresh	Delhi	9156768971	18
2	Ramesh	Gurgaon	9652431543	18

4. ROLLBACK: If any error occurs with any of the SQL grouped statements, all changes need to be aborted. The process of reversing changes is called **rollback**. This command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.

Syntax:

ROLLBACK;

Example:

From the above example **Sample table1**,

Delete those records from the table which have age = 20 and then ROLLBACK the changes in the database.

Queries:

DELETE FROM Student WHERE AGE = 20;

ROLLBACK;

Output:

Student				
Rol_No	**Name**	**Address**	**Phone**	**Age**
1	Ram	Delhi	9455123451	18
2	Ramesh	Gurgaon	9652431543	18
3	Sujit	Rohtak	9156253131	20
4	Suresh	Delhi	9156768971	18
3	Sujit	Rohtak	9156253131	20
2	Ramesh	Gurgaon	9652431543	18

SAVEPOINT: creates points within the groups of transactions in which to ROLLBACK.

A SAVEPOINT is a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction.

Syntax for Savepoint command:

SAVEPOINT SAVEPOINT_NAME;

This command is used only in the creation of SAVEPOINT among all the transactions.

In general ROLLBACK is used to undo a group of transactions.

Syntax for rolling back to Savepoint command:

ROLLBACK TO SAVEPOINT_NAME;

We can ROLLBACK to any SAVEPOINT at any time to return the appropriate data to its original state.

Example:

From the above example **Sample table1**,

Delete those records from the table which have age = 20 and then ROLLBACK the changes in the database by keeping Savepoints.

Queries:

```
SAVEPOINT SP1;                          //Savepoint created.
  DELETE FROM Student WHERE AGE = 20;   //deleted
  SAVEPOINT SP2;                        //Savepoint created.
```

Here SP1 is first SAVEPOINT created before deletion.In this example one deletion have taken place.

After deletion again SAVEPOINT SP2 is created.

Output:

Rol_No	Name	Address	Phone	Age
1	Ram	Delhi	9455123451	18
2	Ramesh	Gurgaon	9652431543	18
4	Suresh	Delhi	9156768971	18
2	Ramesh	Gurgaon	9652431543	18

Deletion have been taken place, let us assume that you have changed your mind and decided to ROLLBACK to the SAVEPOINT that you identified as SP1 which is before deletion.

Deletion is undone by this statement ,

ROLLBACK TO SP1; //Rollback completed.

Student				
Rol_No	Name	Address	Phone	Age
1	Ram	Delhi	9455123451	18
2	Ramesh	Gurgaon	9652431543	18
3	Sujit	Rohtak	9156253131	20
4	Suresh	Delhi	9156768971	18
3	Sujit	Rohtak	9156253131	20
2	Ramesh	Gurgaon	9652431543	18

RELEASE SAVEPOINT:- This command is used to remove a SAVEPOINT that you have created.

Syntax:

RELEASE SAVEPOINT SAVEPOINT_NAME

Once a SAVEPOINT has been released, you can no longer use the ROLLBACK command to undo transactions performed since the last SAVEPOINT.

It is used to initiate a database transaction and used to specify characteristics of the transaction that follows.

2. ISOLATION LEVEL

- Degree of interference
- An isolation levels mechanism is used to isolate each transaction in a multi-user environment
- **Dirty Reads**: This situation occurs when transactions read data that has not been committed.
- **Nonrepeatable Reads**: This situation occurs when a transaction reads the same query multiple times and results are not the same each time
- **Phantoms**: This situation occurs when a row of data matches the first time but does not match subsequent times

Types

Higher isolation level (Repeatable read)

– Less interference

– Lower concurrency

– All schedules are serializable

Lower isolation level (cursor stability)

- More interference
- Higher concurrency
- Not a serializable

One special problem that can occur if transaction operates at less than the maximum isolation level (i.e) less then repeatable read level is called phantom problem.